

Visualisierung laufender Prozesse

Thomas Bauer

DaimlerChrysler Research & Technology
Abt. REI/ID, Postfach 2360, D-89013 Ulm
thomas.tb.bauer@daimlerchrysler.com

Zusammenfassung: Die Visualisierung der Struktur laufender Prozesse inkl. Modellierungsinformation, aktuellem Ausführungszustand und Applikationsdaten ist eine für alle an einer prozessorientierten Applikation Beteiligten sehr wichtige Funktionalität. Leider ist eine solche heutzutage bei den meisten Applikationen nicht realisiert. Ein entscheidender Grund hierfür ist die mangelnde Unterstützung für Visualisierungen durch existierende Werkzeuge. Dies gilt auch dann, wenn ein Workflow-Management-System (WfMS) zur Realisierung der Applikation verwendet wird. In diesem Beitrag stellen wir zwei Ansätze zur Prozessvisualisierung vor und vergleichen sie. Diese haben die positiven Eigenschaften, dass mit relativ geringem Aufwand eine Prozessvisualisierung realisiert werden kann, die auch für nicht IT-geschulte Endbenutzer geeignet ist.

1 Einleitung

In prozessorientierten Applikationen ist die graphische Visualisierung von Prozessen eine sehr nützliche Funktionalität zur Information der Prozessbeteiligten über den aktuellen Ausführungszustand und über das Zusammenspiel der verschiedenen Rollen. Es dient außerdem zum Controlling der Prozessbearbeitung durch das Management und Prozessverantwortliche. Leider ist die Prozessvisualisierung bei heutigen Applikationen nicht ausgeprägt, obwohl dieser Aspekt für die Anwender sehr wichtig ist [Ba04, BS03]. Ein Grund hierfür ist, dass WfMS [Ba01, LR00, JBS97] zwar für die Erstellung von prozessorientierten Applikationen sehr nützlich sind, die Prozessvisualisierung aber vernachlässigen. Deshalb wird in diesem Beitrag untersucht, welche Möglichkeiten es gibt, um dennoch eine Prozessvisualisierung zu realisieren. Hierbei sind vorrangige Ziele, mit begrenztem Umsetzungsaufwand auszukommen und eine für den Endbenutzer möglichst leicht verständliche Darstellung zu erreichen, die aber alle relevante Information enthält.

Nach einer kurzen Diskussion verwandter Themen wird in Abschnitt 3 ein Ansatz vorgestellt, bei dem eine geeignete Prozessvisualisierung auf Basis zuvor getroffener Festlegungen automatisch generiert wird. In Abschnitt 4 wird aufgezeigt, wie eine zuvor „gemalte“ Darstellung der Prozessvorlage automatisch so erweitert werden kann, dass auch aktuelle Laufzeitinformation des Workflows dargestellt werden kann. Der Beitrag schließt mit einer Zusammenfassung und einem Vergleich der beiden Vorgehensweisen.

2 Diskussion

Im Folgenden wird untersucht, inwieweit verschiedene Systemtypen zur benutzergerechten Visualisierung laufender Prozessinstanzen geeignet sind.

Wie schon erwähnt, ist die Prozessvisualisierung existierender WfMS unzureichend. Abb. 1 zeigt eine Prozessvisualisierung des bezüglich dieser Funktionalität relativ mächtigen WfMS Lotus Workflow [Lo00]. Dieses hat, ebenso wie die meisten anderen WfMS, die folgenden Schwachstellen:

- Es kann nur direkt diejenige Sicht auf einen Workflow dargestellt werden, die bei der Modellierung erzeugt wurde. Dies ist aber eher ein technischer Blickwinkel, d.h. er beinhaltet auch automatische Aktivitäten oder durch das Metamodell bedingte Subworkflows. Dadurch wird die Darstellung für die Endanwender zu unübersichtlich.
- Es können nur WfMS-interne Attribute wie Bearbeiter oder Startzeiten dargestellt werden. Es ist meist nicht möglich, Anwendungsdaten zu visualisieren. Diese sind allerdings für die Endbenutzer höchst relevant.
- Die graphische Darstellung ist häufig sehr unübersichtlich. Normalerweise kann diese auch nicht um Zusatzinformationen angereichert werden, welche die Übersichtlichkeit erhöhen würden (z.B. Texte, Linien, Graphiken, farbige Hintergründe).

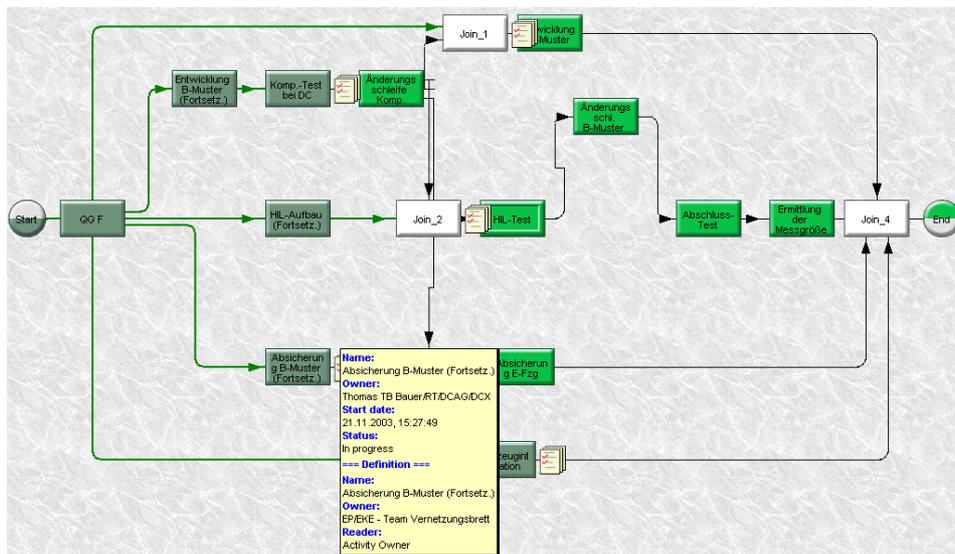


Abbildung 1: Visualisierung einer Workflow-Instanz mit Lotus Workflow

Geschäftsprozessmodellierungswerkzeuge [Sch98] wie ARIS oder Bonapart erlauben eine allgemein verständliche Darstellung von Geschäftsprozessen. Da aber ausschließlich die Modellierung im Fokus liegt, ist es mit ihnen nicht möglich, aktuelle Ausführungszustände oder Applikationsdaten zu visualisieren.

Projektmanagementsysteme wie Microsoft Project oder RPlan von Actano erlauben die Darstellung sowohl eines Ablaufmodells als auch des aktuellen Bearbeitungszustands, aber nicht von Applikationsdaten. Außerdem ermöglichen sie nur die Darstellung von sehr speziellen Abläufen, nämlich Projektplänen, welche keine Schleifen oder bedingten Verzeigungen enthalten können. Schließlich ist die Visualisierung meist recht unübersichtlich, wobei allerdings die zeitbezogene Darstellung oft als intuitiv empfunden wird.

Werkzeuge zum Process Performance Management wie IBM WBI Monitor Business Dashboard [IBM03] oder der Process Performance Manager [IDS02] von IDS-Scheer erlauben zwar die Darstellung von Ausführungsinformation, allerdings auf einer aggregierten Ebene für das Management. Deshalb wird über Kennzahlen und Diagramme eingestiegen und eine Verfeinerung bis auf Prozessebene ist aufwendig und die resultierende Darstellung recht unübersichtlich.

3 Generierung einer geeigneten graphischen Darstellung

Eine für eine große Zielgruppe verwendbare Prozessvisualisierung sollte zumindest die folgenden Anforderungen erfüllen:

- Einfach aufrufbare und übersichtliche Darstellung (durch Beschriftungen, Farben, Graphikelemente, etc.). Nicht relevante oder verwirrende Objekte müssen ausblendbar sein (z.B. automatisch ausgeführte Join-Aktivitäten).
- Darstellung aller relevanter Struktur-, Modell-, Zustands- und Ausführungsinformationen, aus beliebigen Quellsystemen.
- Personalisierbare und leicht anpassbare (ggf. dynamisch) Darstellung, je nach Bedarf und Vorkenntnissen des Benutzers.
- Darstellbarkeit unterschiedlicher Sichten (z.B. ausgehend von Prozess, Organisationseinheiten, IT-Systemen oder Kennzahlen für das Management).
- Darstellung von (Gesamt-) Geschäftsprozessen, die von unterschiedlichen IT-Systemen (bzw. teilweise manuell) gesteuert werden.

Um diese Anforderungen erfüllen zu können, wurde der folgende Ansatz prototypisch in einem Anwendungsprojekt bei DaimlerChrysler realisiert: Die Workflows werden durch das WfMS Lotus Workflow [Lo00] gesteuert, welches auch den Export der Workflow-Vorlagen in einem XML-Format ermöglicht. Diese Vorlagen, ein selbst definiertes Visualisierungsmodell und Anwendungsdaten, die teilweise außerhalb des WfMS gespeichert sind, stellen die Datenquellen für eine als Servlet selbst realisierte Visualisierungskomponente dar (siehe Abb. 2). Die Verbindung der Anwendungsdaten mit der Workflow-Instanz wird durch einen „Datenzugriffsabschnitt“ im Visualisierungsmodell definiert, wohingegen das restliche Visualisierungsmodell die Generierung der graphischen Darstellung des Workflows steuert. Generiert wird dabei SVG (Scalable Vector Graphics [Ei02]), ein auf XML basierendes Graphikformat, das von Web-Browsern dargestellt werden kann.

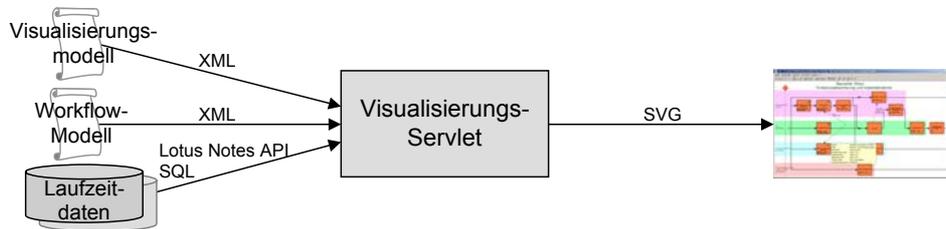


Abbildung 2: Ablauf bei der Generierung einer Prozessinstanz-Visualisierung

Der Vorteil dieses Ansatzes ist, dass es durch geeignete Festlegungen im Visualisierungsmodell prinzipiell möglich ist, jede gewünschte graphische Darstellung zu generieren. So können, wie in Abb. 3 dargestellt, Texte, Pfeile, Rechtecke und Graphikobjekte beliebig dazu verwendet werden, eine endbenutzergerechte Visualisierung zu erzeugen. Selbstverständlich ist es auch möglich, die Darstellung der Aktivitäten und Kanten zu beeinflussen (dargestellte Daten, Größe, Farbe), sowie sie teilweise ganz auszublenden bzw. ihre Anordnung auf der Zeichenfläche zu verändern. Außerdem kann durch den Datenzugriffsabschnitt jede Art von Laufzeitdaten einbezogen werden, unabhängig davon, ob diese im WfMS oder in einer externen Datenquelle gespeichert sind.

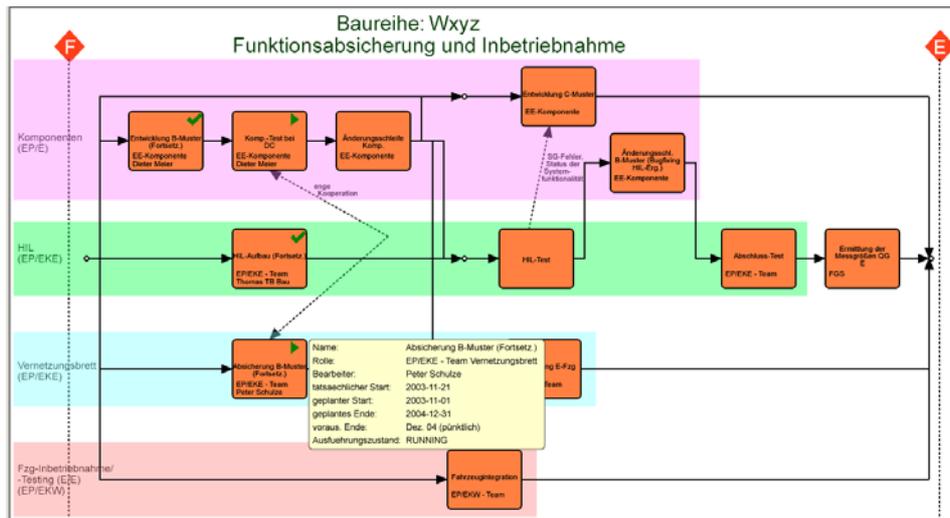


Abbildung 3: Automatisch generierte graphische Visualisierung einer Workflow-Instanz

Eine entsprechende Visualisierungsfunktionalität sollte zukünftig idealerweise in die Modellierungskomponenten der WfMS übernommen werden, da sie (mit Ausnahme des Visualisierungsmodells selbst) anwendungsunabhängig ist. Bei der Workflow-Modellierung könnten geeignete Sichten für unterschiedliche Benutzergruppen festgelegt werden, die dann jeweils in einem Visualisierungsmodell gespeichert werden. Diese werden später verwendet, um eine geeignete Prozessvisualisierung zu generieren, wobei die in diesem Abschnitt vorgestellte Vorgehensweise zur Anwendung kommt.

4 Anpassung vorgefertigter Prozessdarstellungen

Eine alternative Vorgehensweise, die in einem aktuell laufenden Projekt umgesetzt wird, ist, die Visualisierung des Workflow-Modells manuell mit einem Editor zu erstellen. Diese Darstellung enthält dann schon genau diejenigen Gestaltungselemente, die für eine benutzergerechte Visualisierung erforderlich sind. Sie muss dann vor jedem Anzeigen nur noch geringfügig angepasst werden, um auch Laufzeitdaten des Workflows darstellen zu können. Hierfür gibt es zwei generelle Ansätze, die im Folgenden beschrieben werden.

1. In eine Graphik (z.B. SVG) kann Programmcode (JavaScript bei SVG) eingebaut werden, dessen Ausführung wiederum SVG erzeugt. Dieser Mechanismus wird genutzt, um bei der Anzeige der SVG-Graphik auf die Datenbanken mit der Laufzeitinformation (aktueller Zustand, Applikationsdaten) zuzugreifen. Aus diesen Daten werden dann SVG-Objekte zur Darstellung des aktuellen Status generiert, die vom Browser ebenfalls angezeigt werden.
2. Wird eine Visualisierung angefordert, so liest das entsprechende Servlet ein „SVG-Template“ ein, welches noch Platzhalter für darzustellende Laufzeitinformationen enthält. Nach Zugriffen auf die WfMS- und Applikationsdatenbanken werden diese dann durch die entsprechenden SVG-Objekte ersetzt, bevor der SVG-Code zur Darstellung an den Browser weitergegeben wird.

Prinzipiell bieten beide Ansätze dieselben Möglichkeiten. Allerdings werden beim 2. Ansatz einige Schwierigkeiten vermieden: So muss nicht durch den von einem Browser ausgeführten JavaScript-Code auf die Laufzeitdatenbanken zugegriffen werden, was zu Problemen mit Firewalls führen kann. Stattdessen erfolgt dieser Zugriff stets durch den Servlet-Server. Außerdem kann bei der Servlet-Programmierung der volle Sprachumfang von Java genutzt werden und es müssen nicht unterschiedliche JavaScript-Dialekte unterschiedlicher Web-Server berücksichtigt werden. Deshalb ist der 2. Ansatz normalerweise besser geeignet.

5 Zusammenfassung und Vergleich der Ansätze

Die Visualisierung von Prozessinstanzen ist für die Benutzer eines prozessorientierten Applikationssystems sehr wichtig, wird aber zumeist nicht realisiert. Der Hauptgrund hierfür ist der erforderliche Aufwand und die niedrige Priorität einer Prozessvisualisierung gegenüber der reinen Prozesssteuerung. Existierende Werkzeuge sind auch nicht geeignet, den Implementierungsaufwand zu reduzieren, da sie andere Ziele verfolgen. Selbst bei Verwendung eines WfMS muss eine benutzergerechte Visualisierung selbst implementiert werden, obwohl ein WfMS eigentlich über alle für eine Visualisierung benötigten Informationen verfügt.

In diesem Beitrag haben wir in Anwendungsprojekten ermittelte Anforderungen vorgestellt, sowie zwei generelle Möglichkeiten, um eine Prozessvisualisierung im SVG-

Format zu realisieren. Deren Eigenschaften sollen nun verglichen werden: Die Generierung von SVG-Graphiken (Abschnitt 3) erfordert einen höheren Implementierungsaufwand und ermöglicht weniger Darstellungsmöglichkeiten als das direkte „Malen“ einer SVG-Graphik (Abschnitt 4). Außerdem kann der Generierungsansatz nur verwendet werden, wenn der Prozess von einem WfMS gesteuert wird. So betrachtet ist eigentlich immer der Ansatz aus Abschnitt 4 vorzuziehen. Dieser kann sogar bei systemübergreifenden Prozessen eingesetzt werden. Allerdings gibt es in diesem Zusammenhang noch viele ungelöste Fragestellungen und damit Forschungsbedarf.

Der Ansatz aus Abschnitt 3 hat aber bei einer (mehr oder minder engen) Integration mit dem Modellierungswerkzeug des WfMS auch seine Berechtigung, da er dann eine gewisse Flexibilität bzgl. der Workflow-Modelle ermöglicht: Wird eine Workflow-Vorlage verändert, so muss sie lediglich exportiert werden, um eine Visualisierung auf Basis der neuen Vorlage zu ermöglichen. Dies gilt insbesondere, wenn wie bei Lotus Workflow auch Layoutinformation mit exportiert wird. Hingegen müsste eine gemalte SVG-Graphik erst manuell angepasst werden, bevor eine Visualisierung basierend auf einer veränderten Workflow-Vorlage möglich wird.

Literaturverzeichnis

- [Ba01] Bauer, T.: Effiziente Realisierung unternehmensweiter Workflow-Management-Systeme. Dissertation Universität Ulm, Tenea-Verlag, 2001.
- [Ba04] Bauer, T.: Integration heterogener Applikationstypen durch Workflow-Management-Technologie, In: Proc. Workshop on Enterprise Application Integration, Oldenburg, 2004; S. 39-47.
- [BS03] Bauer, T.; Siebert, R.: Requirements and Methods for the Coupling of Project and Workflow Management Systems. In: Proc. ProSTEP iViP Conf. on Advances in Collaborative Product Creation, Dresden. 2003; S. 173-185.
- [Ei02] Eisenberg, J.D.: SVG Essentials, O'Reilly & Associates, 2002.
- [IBM03] IBM: WebSphere Business Integration Monitor, Version 4.2.4, Specification Sheet, 2003.
- [IDS02] IDS Scheer: ARIS Process Performance Manager: Messen, Analysieren und Optimieren Sie Ihre Geschäftsprozess-Performance, 2002.
- [JBS97] Jablonski, S.; Böhm, M.; Schulze, W.: Workflow-Management: Entwicklung von Anwendungen und Systemen, dpunkt-Verlag, 1997.
- [Lo00] Lotus: Domino Workflow Application Development, 2000.
- [LR00] Leymann, F.; Roller, D.: Production Workflow - Concepts and Techniques, Prentice Hall, 2000.
- [Sch98] Scheer, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen. Springer-Verlag, 1998.